Ensimag 2ème année
SLE 2014 – 2015

Projet Bases de Données et Conception Objet

1 – Introduction

Ce projet a pour but de mettre en œuvre vos compétences en systèmes de gestion de bases de données relationnelles et en conception objet. Il vient en application du cours de « Principes des systèmes de gestion de bases de données » (PSGBD) et du cours d' « Analyse, conception et validation des logiciels » (ACVL). Vous devez réaliser l'analyse, la conception et la programmation complète d'une application de gestion de batailles navales utilisant une base de données. Vous devez notamment concevoir et implanter un schéma relationnel. Le programme doit être écrit en *Java*, et les accès à la base de données effectués grâce l'API *JDBC*.

Le projet est à faire en équipe de 5 élèves donnera lieu à la remise d'un rapport, ainsi qu'à une soutenance comprenant, entre autres, une démonstration. L'évaluation concerne l'analyse, la conception et l'implémentation, à la fois sur les aspects bases de données et sur la partie conception orientée objets.

2 – Description de l'application

L'application que vous devez réaliser permettra la gestion d'une salle de jeux virtuelle organisant des batailles navales, à l'aide d'une interface graphique. Elle consistera en plusieurs aspects : i) la gestion des inscriptions des différents participants à la salle de jeux ; ii) l'organisation de parties de bataille navale en sélectionnant aléatoirement les joueurs ; iii) le déroulement en parallèle de parties de bataille navale (enchainement des tours de jeu, respect des règles, *etc.*) ; iv) l'observation par des inscrits de parties en cours, avec possibilité de rejouer l'historique des coups.

2.1 –La salle de jeux

La salle de jeux a deux fonctions : la gestion des joueurs et la gestion des parties de bataille navale.

2.1.1 – Inscription des joueurs

Quiconque souhaitant jouer doit s'inscrire à salle de jeux. Cette personne, une fois inscrite, devient un joueur de la salle de jeux : elle peut alors utiliser l'ensemble des fonctions proposées.

Un joueur est identifié par son pseudo et décrit par son nom, son prénom, son email, son adresse (numéro, rue, code postal et ville) et sa date de naissance. Ces informations lui sont demandées lors de son inscription.

2.1.2 – Organisation des parties

Un joueur de la salle de jeux peut demander à lancer une partie. L'application lui assigne alors un adversaire de la façon suivante : l'adversaire est sélectionné de façon pseudo-aléatoire parmi les autres joueurs de la

salle qui ne sont pas déjà en train de jouer, les joueurs ayant joué le moins de parties ayant plus de chance d'être sélectionnés en premier.

Indication : Les parties ont un identifiant unique et sont décrites par leur date de démarrage, par les joueurs participants et par le vainqueur (pour les parties terminées).

2.2 – Règles du jeu et déroulement d'une partie

Une partie de bataille navale se déroule sur deux espaces maritimes séparés et indépendants. Chaque joueur voit son espace, une grille de 10x10 cases, et uniquement son espace. De ce fait, il n'y a jamais de collision entre bateaux adverses.

Au début de chaque partie, chacun des joueurs dispose d'un destroyer (bateau occupant 3 cases consécutives en mer) et peut choisir d'avoir un ou deux escorteurs (bateau occupant 2 cases consécutives en mer). Chaque bateau, identifié de façon unique pour une partie, doit être placé sur la grille. Bien sûr, deux bateaux ne peuvent occuper la même case sur la grille. Un bateau peut être placé horizontalement ou verticalement. Seul le joueur connait son choix en nombre et placement de bateaux.

Une fois tous les bateaux placés, la partie peut commencer. Les joueurs jouent à tour de rôle en commençant par celui qui a été sélectionné en premier (joueur 1).

Lors d'un tour de jeu, un joueur réalise des actions sur chacun de ses bateaux. Un bateau peut réaliser autant d'actions qu'il a de cases au départ. Il perd une action à chaque fois qu'il est touché, quel que soit l'endroit (un destroyer intact peut réaliser 3 actions par tour, un destroyer touché une fois ne peut en réaliser que deux par tour, *etc.*)

Les actions possibles sont :

1. Un déplacement : faire un quart de tour à gauche ou à droite (la case arrière du bateau servant de pivot), avancer ou reculer d'une case, le tout dans les limites de la mer (10x10).
   Indication : la contrainte de validité des mouvements devra être implantée au niveau du SGBD. Il est important de représenter un bateau par son point de pivot, sa taille et son orientation.
2. Un tir : un bateau peut tirer n'importe où dans l'espace adverse, quel que soit son emplacement.

Une fois les actions de tous les bateaux du joueur sélectionnées, elles sont exécutées. A ce moment-là l'efficacité des tirs est déterminée et le joueur adverse est informé des coordonnées des tirs et des mouvements des bateaux.

La main passe ensuite au joueur suivant qui peut jouer à son tour. Pour cela, le joueur dispose d'un bouton « Rafraîchir » qui permet de passer au tour suivant si son adversaire a joué.

Un bateau touché autant de fois qu'il a de cases est coulé, même si c'est à chaque fois au même endroit. *Indication : L'état des bateaux doit donc être mémorisé.*

La partie s'arrête lorsqu'un des joueurs n'a plus de bateau.

2.3 – Observation et rejeu de parties en cours ou terminées.

Un observateur est un joueur de la salle mais qui ne participe à aucune partie. Il sélectionne une des parties (en cours ou terminée) qu'il veut observer. Il voit alors les deux espaces de mer et a la possibilité de voir

l'état courant de la partie. Il peut demander à actualiser cet état courant (à l'aide d'un bouton « Coup suivant », par exemple) pour voir si un coup a été joué. Il peut aussi rejouer la partie depuis le début.

Indication : il faut donc conserver l'historique des actions jouées pour chacune des parties.

3 – Travail à réaliser (15h encadrées)

3.1 – Analyse

Cette étape consiste à analyser le problème posé.

a) **Réaliser *l'analyse dynamique***, en identifiant les différents cas d'utilisation. Faire un diagramme de cas d'utilisation. Documenter le fonctionnement global du système par un ou plusieurs diagrammes états/transitions pertinents. Documenter les cas d'utilisation par des explications, *si nécessaire* illustrées par un ou plusieurs diagrammes de séquence.
b) **Réaliser *l'analyse statique***. Inventorier les données élémentaires, identifier les dépendances fonctionnelles et les contraintes (invariants) entre les données persistantes.

3.1 – Conception de bases de données

a) ***Élaboration du schéma conceptuel***. Construire, à partir de l'analyse statique un schéma Entités/Associations (ce schéma correspond à un diagramme de classe d'analyse). Expliquer ce schéma (notamment les points difficiles de sa construction). Préciser les contraintes d'intégrité non représentées dans le schéma Entités/ Associations.
b) ***Conception de la base de données***. Traduire le schéma Entités/Associations en relationnel ; écrire les tables et les contraintes. Préciser et justifier la forme normale des relations, les contraintes d'intégrité implantées, ainsi que les contraintes ne pouvant être implantées en SQL2 (et qui sont donc à vérifier dans le code applicatif Java).
c) ***Analyse des accès à la base de données***. Écrire les requêtes nécessaires aux différentes fonctionnalités à implanter. Délimiter les transactions permettant d'implanter les fonctionnalités tout en préservant la cohérence globale des données.

3.3 – Conception de l'application

a) ***Architecture de l'application***. Proposer une architecture pour l'application. Préciser en particulier l'interface utilisateur ainsi que les liens avec la base de données. Proposer un diagramme de collaboration ou un diagramme de séquences illustrant un scénario typique pour :
   i. une action de l'utilisateur sur l'interface.
   ii. le déclenchement et l'exécution d'une requête depuis l'application.
b) ***Conception objet***. Faire un diagramme de classes logicielles.

3.4 – Implémentation

a) ***Instancier la base de données***. Elle doit être créée sur le serveur Oracle 12c disponible sur le serveur *ensioracle1*.
b) ***Programmer l'application en Java***. L'accès à la base de données se fera avec l'API JDBC.

3.5 – Bilan

Rédiger un bilan de ce projet (répartition des tâches, difficultés rencontrées, leçons retenues, *etc.*). Pour vous aider dans cette tâche, il vous est demandé de maintenir **un journal** de projet dans lequel vous

préciserez (en quelques lignes) ce qui a été réalisé au cours de chaque séance encadrée de projet (ou en dehors de ces séances). Ce journal devra être déposé sur l'application *Teide* à la fin de chaque séance encadrée (déposé, mais non validé) et pourra être consulté à tout moment par vos encadrants tout au long du projet.

La ***documentation finale*** à rendre comporte quatre parties : l'analyse, la conception et l'implantation de la base de données, la conception de l'application, et le bilan.

4 – Soutenance (Mercredi 29 avril 2015 à partir de 14h)

Votre projet donne lieu à une soutenance de 20 minutes (inscription sur les créneaux horaires *via* l'application *Teide*), suivie de 10 minutes de questions, durant laquelle vous devrez :

- présenter votre modélisation entités-associations et son implantation en relationnel
- présenter votre architecture
- présenter le diagramme de classes logicielles
- faire une démonstration des fonctionnalités ; prévoir un scénario montrant le bon fonctionnement des transactions
- faire un bilan du déroulement du projet

L'objectif est de vendre **votre produit** (en montrant qu'il fait bien ce qu'on attend) **et votre équipe de projet** (en montrant que le produit a été bien construit).

Note : Il est très fortement recommandé de consulter régulièrement les pages du kiosk consacrées à ce projet.